From Customer Requirements to PIM: necessity and reality

Ana Belen Garcia, Jason Mansell, David Sellier
European Software Institute,
Parque Tecnológico #204,
E-48170 ZAMUDIO,
Bizkaia, Spain

Abstract

Model Driven Architecture (MDA) is a recent approach to software development that specifically aims at uniquely identifying two main parts of any system, the platform independent model (PIM) and the platform specific model (PSM). Taking full advantage of this potential requires adequate planning and management of the system functionalities and customer requirements.

Currently we can identify two major issues which are being investigated and in which many initiatives are taking place. One of these issues is the one related to PSM in which there are many initiatives regarding the definition of standard PSM for specific platforms, this initiative is clearly identified within the responsibility of the platform developers and standardisation organisations, which must provide us, software engineers, with standard PSMs. The other issue is related to the automatic code generation that some UML tool providers offer and which is continuously improving.

A key issue in the MDA approach which requires much work is the customer requirements to PIM implementation and automation process.

In this paper, we describe an integrated approach that has been identified and is being developed, improved and validated within the MASTER project (IST-2001-34600). The approach fully covers the definition of customer requirements for a product line and the mapping activities from customer requirements to PIM. This approach will make it possible to automate the generation of code based on customer requirements. This automation process requires the use of the Product Line Engineering concepts. The product line engineering is an approach to software development that specifically aims at exploiting commonalities and variabilities among functionally overlapping systems within a specific domain in terms of large-scale reuse.

The whole approach is based on XML technology, however this underlying technology will be transparent for the users of the approach. In addition, this customer requirement to PIM mapping technique is already being prototyped and demonstrated.

Overview of the approach: from customer requirements to the PIM

Every system defined using the MDA approach is formalised in at least two groups of UML models, one for the PIM, which will contain the business logic of the system and at least one PSM which will contain the platform implementation knowledge of the system.

We consider that it is possible to automate the derivation process from customer requirements to code, but this is only possible through a Product line approach since there is a need for a stable domain in which a generic architecture of the PIM for that system has been defined and developed.

On the other hand, for single systems that do not belong to a product line, the possibility of automation is limited to the last development phases, this is, from the PIM (manually produced) to the code. There is not a way to automate the production of PIM for unique systems, although some automation for refinement between different levels of abstraction within the PIM is still possible.

Back to the product line context, the domain PIM contains the common and variable parts of the system which should be taken into account when defining a specific PIM for a specific customer. Therefore there is a need to identify what parts of the PIM are common to all the possible final systems and which parts are dependent of the user requirements. This activity has been defined as a domain analysis in which the common and variable parts of the product line are identified and specified conforming a Domain Model of the product line.

The commonalities identified in a domain analysis will form the basic architecture of the generic domain PIM, where as the variabilities will determine points of variation within that generic PIM, These points of variation will be resolved in specific values and will determine a specific system PIM for specific user requirements. This idea is captured in the concept of Flexible PIM. A Flexible PIM is a PIM which contains both the commonality and variability of the business logic of the system.

Another important concept within this approach is the Decision Model which will contain all the decisions that have to be taken by customers and that are implemented in the Flexible PIM in order to obtain a user requirements based PIM, for the final system.

How to create a Decision Model

The result of a domain analysis basically consists on a number of commonalities and variabilities that will have to be implemented within the Flexible PIM. The commonalities conform the common part of the architecture of the flexible PIM while the variabilities are defined in a decision model that represents all possible user requirements defined during the domain analysis and the set of rules and constrains associated with them.

We have defined a Meta-Model for the Decision Model, which is used to specify decision models of product lines and to validate that all the decision models implemented adjust to the same set of rules.

This Meta-Model specifies:

1. The decision model structure, which determines how the decisions should be structured.

2. The decision, which determines how a decision is defined in a complete way. This is the result of the study of the characteristics and the types of decisions that appear in a decision model.

3. The dependencies among decisions, which determines relationships and constraints between decisions. Dependencies strongly condition the way to resolve the decisions of a Decision Model and also impact the implementation of the variation points.

Since the way to specify the structure of the decisions, the decisions themselves and the dependency among decisions are well known and well defined in this meta-model, some tools can be used in order to support the building of the decision model. Currently, we have developed a prototype of such a tool. In addition, by specifying the decision model built on this Meta-Model it makes it possible to interpret the decision model.

Technology used to represent the Decision Model

Both the Meta Model Decision Model and the Decision Model are represented with the XML technology, more precisely with an XML Schema.

The decision model represents all possible user requirements defined during the domain analysis and the set of rules and constraints associated with them.

The decision model can be shown as a tree structure where decisions can be grouped into a set of decisions. A decision is defined by a set of elements that identify the decision and a decision is always represented as leafs of the tree as shown in the following picture:

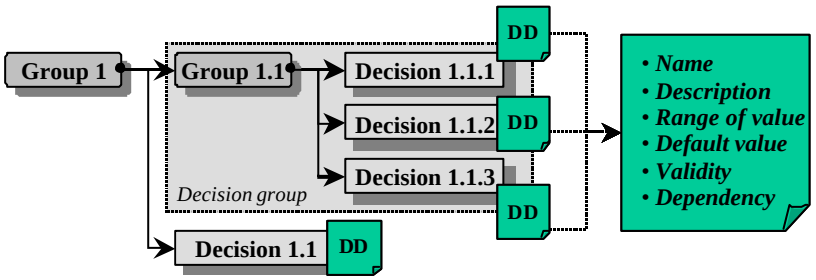

Figure 1 – Decision Structure

The decision model defines different types of decisions, which have associated an implementation mapping in the XML Schema.

The XML Schema is used to map the decision model so as to represent the decision structure, to specify each decision as well as the variability specifications of the domain. The decision model is the sequence of decisions that the user should make.

The next example shows the implementation of a decision type in which the decision can adopt a specific value within a range of values:

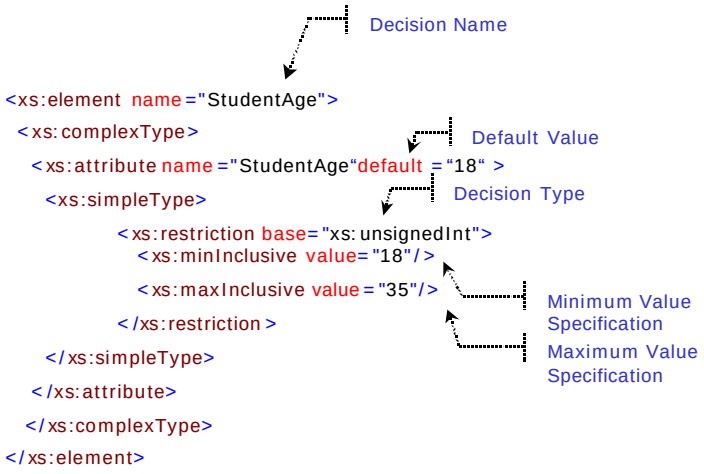

Figure 2 – Minimum and Maximum Value Specification within the XML Schema

The choice of XML technology is based on a reasoned analysis:

- On the one hand, XML allows the definition of all the existing types of decisions, allows structuring decisions in a Decision Model and provides the programmatic functionality (through XSL) to implement dependency.

- On the other hand, the variation points associated to each decision have to be implemented in the PIM (Flexible PIM), more specifically in the UML models that make up the PIM. The publication of XMI implies that each UML model can be also represented as an XML file. Our approach to design the Flexible PIM is to implement the variation points in the PIM XMI. Thus, we use the same technology both to represent the decision model and to implement the variation points. This provides a powerful mechanism to automate the generation/instantiation of a PIM from the user requirements (from specific decisions).

How to create a Flexible PIM

Flexible PIMs implement application functionalities that depend on the variability-commonality within the domain. The core of the Flexible PIM implements both the common and variable part of the domain. The variability within Flexible PIMs consists of a set of variation points in which values determine the tailoring required for instantiation. Flexible PIMs as well as the operational interfaces provide a set of interfaces (derivation interfaces) to capture the variation point values.

Flexible PIMs implement the variability and the commonality of the domain, while providing interface(s) to capture the variation point values (user requirements). The relation between the product line engineering and Flexible PIMs construction is described in Figure 3.

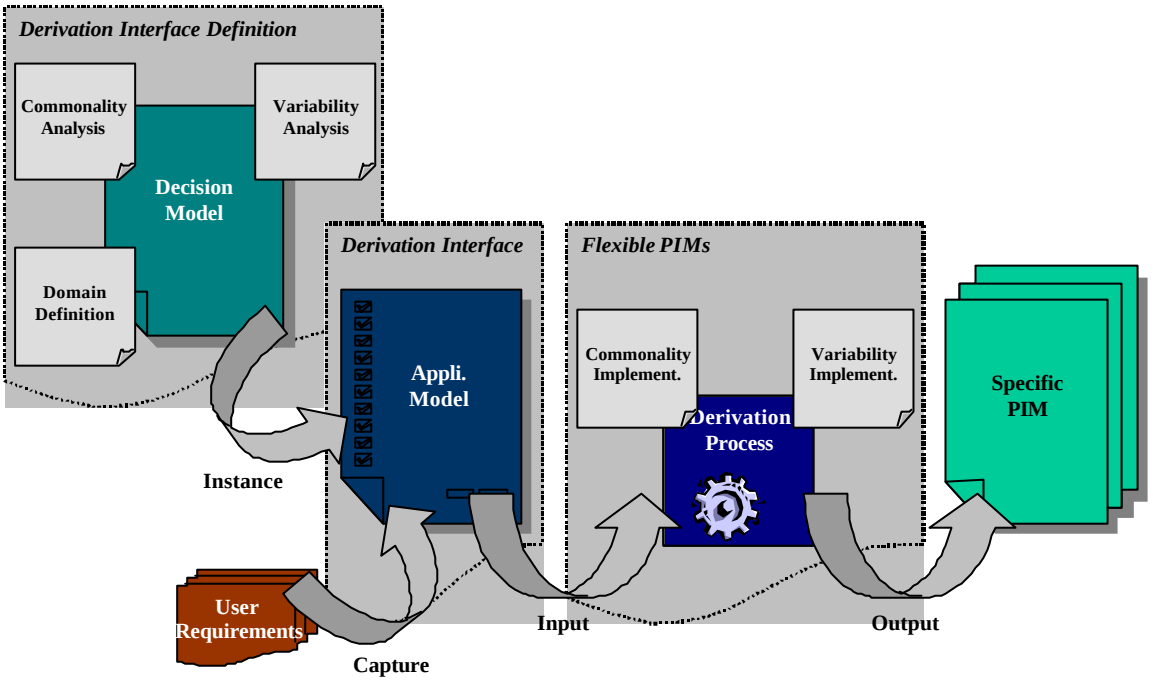

Figure 3 - Product line Engineering and Flexible PIMs

Technology used to build Flexible PIMs

Flexible PIMS are also build using XML technology. The following Figure 4 shows how the XML technology is used to implement the different phases of the product line engineering approach shown in Figure 3.

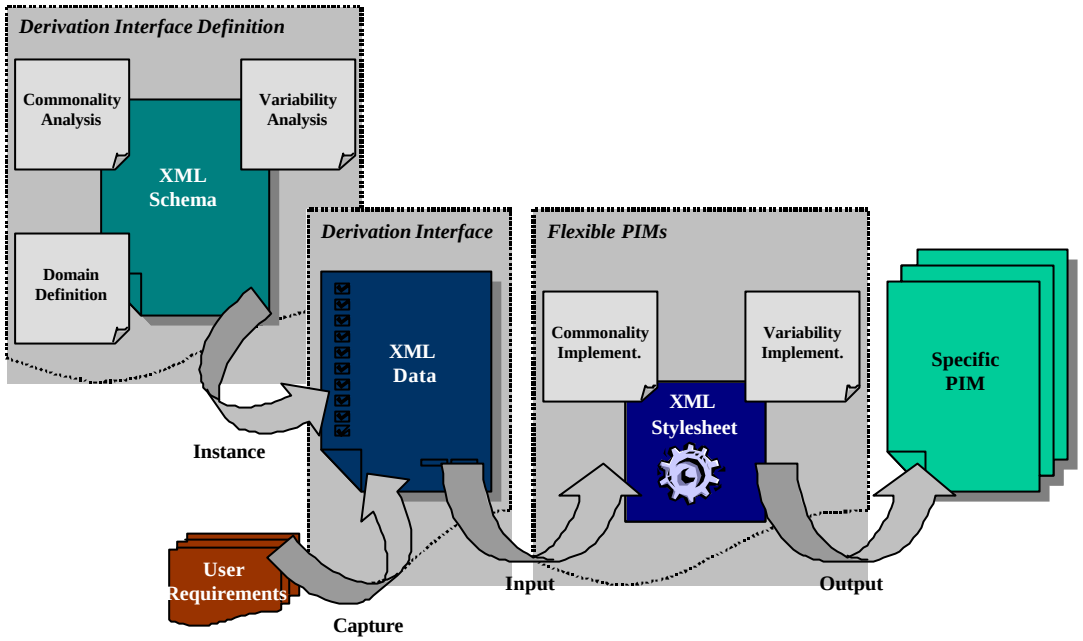

Figure 4 - The XML Technology and Flexible PIMs

The derivation interface of Flexible PIMs is represented using the XML Data document (XML) that captures user requirements. Flexible PIMs are defined and are specified with XML Stylesheet documents (XSLT). XML Stylesheets provide the necessary functionality to implement the variation points.

The domain PIM is implemented by a set of XSL files which generate different XMI files depending on the decisions (customer requirements). These XMI files correspond to an UML PIM definition that corresponds to the user requirements needs and definition.

Conclusion

This paper shows an approach to automate the generation of a system PIM from the system customer requirements. On the one hand, the underlying engineering concepts are MDA and Product line engineering. In the other hand, the representation and automation technique is based on XML technology.

The approach presented in this paper, which is already being prototyped and demonstrated, pursues the automation of system development for systems belonging to a specific product line. This work has to be integrated with other research activities in the field of PIM to PSM and PSM to code mappings. Different aspects of this research, including this paper, are being studied in MASTER project (IST-2001-36400).

The following picture shows the overall approach to automate the production of PIMs from requirements specification.

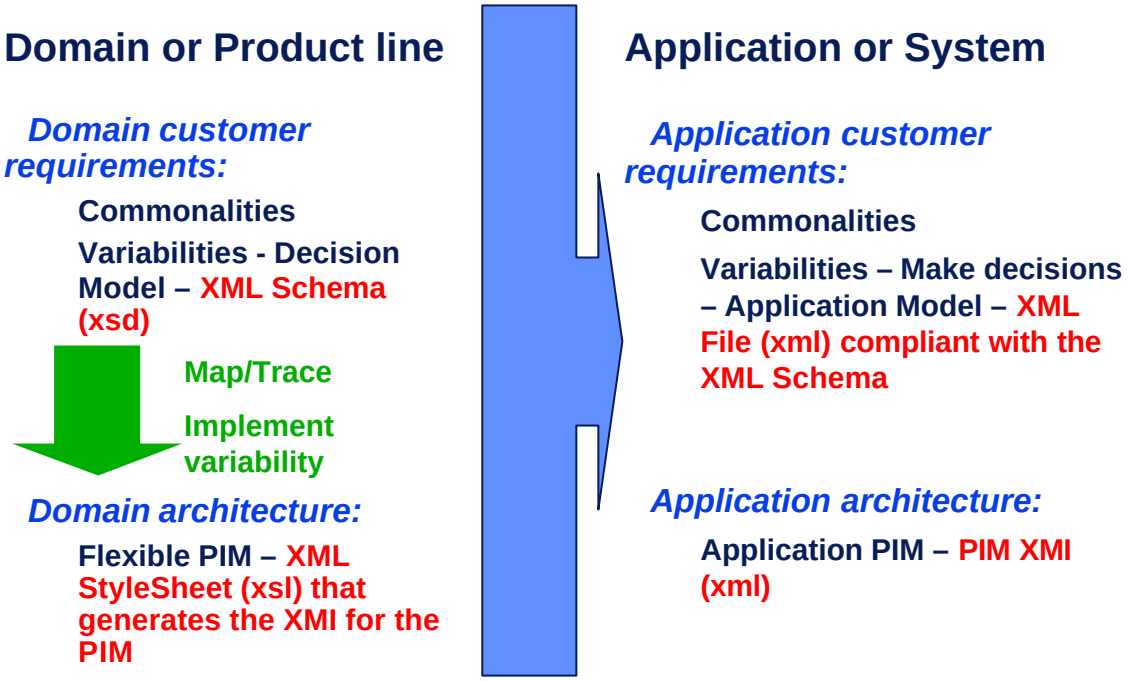

Figure 5 – Overall Approach

Contact persons

Ana Belén García, anabelen@esi.es
Jason Mansell, jason@esi.es
David Sellier, sellier@esi.es

European Software Institute,
Parque Tecnológico #204,
E-48170 ZAMUDIO,
Bizkaia, Spain

Tel.: ++ 34 94 4209519
Fax: ++ 34 94 4209420